

Hemtentamen - Designmönster 2018

Omfattningen på hemtentamen är minst 2000 ord exklusive kodexempel.
(2000 ord motsvarar ungefär fyra A4-sidor).

Fördelningen av textmängden är:

Uppgift 1 - cirka 500 ord

Uppgift 2 - cirka 500 ord

Uppgift 3 - minst 1000 ord

Totalt ska det vara minst 2000 ord (kod räknas inte).

Lycka till!

Hemtentamen - Designmönster 2018

Uppgift 1 - Identifiera patterns i Javas API

max 500 ord

Identifiera förekomster av följande designmönster i Javas API:

- Factory (vilken version som helst)
- Observer
- Strategy
- Decorator
- Singleton

Du ska ge **minst ett exempel** på varje och några rader kod som visar på hur man använder mönstret. Du ska förklara kort **vad för objektorienterade designprinciper som stöds av mönstret** (det vill säga, vad effekten av mönstret är och vad man tjänar på att de valt att implementera det). Några korta meningar räcker för att visa att ni förstått.

Uppgift 2 - Analysera en laboration/inlämningsuppgift du gjort i en tidigare kurs

max 500 ord

Välj en programmeringslaboration eller -inlämningsuppgift du gjort i en tidigare kurs. Vilka mönster använde ni i er lösning? Ge exempel med små kodsnuttar som visar hur ni använde mönstret. Vad (om något) tjänade ni på mönstret? Vilka OO-designprinciper stödde er användning av mönstret? Hade du gjort något annorlunda om du gått designmönsterkursen innan? (Nej är OK som svar, men det är ju bara ett ord).

Uppgift 3 - Utforska ett nytt pattern

Minst 1000 ord (cirka två A4) - kod räknas inte.

Ta reda på hur Visitor Pattern fungerar. Ni får använda internet och böcker. Ge ett *kort* exempel i kod på hur det kan användas (skriv själva, vi känner till de flesta webbsidor med kodexempel ;-). Förklara vilka designprinciper (om några) som stöds av mönstret. Förklara eventuella problem med mönstret (det kanske bryter mot någon eller några designprinciper). Ange källor.

Krav för VG (Kan skippas av dem som siktar på G)

Förklara med egna ord (kodexempel ska vara med men räknas inte som ord i tentans krav på storlek) skillnaden mellan

- Arv

- Komposition
- Decorator

Ge ett exempel på ett system för anställda i ett företag, där systemet hanterar:

- Employee (anställd)
- Chef (anställd som är chef för en avdelning)
- Engineer (anställd som har rollen ingenjör)
- Clerk (anställd som har rollen tjänsteman/administratör)
- Programmer (anställd som har rollen ingenjör och rollen programmerare)

Lönen blir högre för chefer.

En Engineer tjänar mer än en Clerk. En Programmer tjänar mer än andra Engineer.

Systemet ska kunna utökas med nya sorters anställda.

Visa på design-utmaningar eller -fördelar mellan att välja en design med Arv, Komposition eller Decorator. Korta exempel premieras.